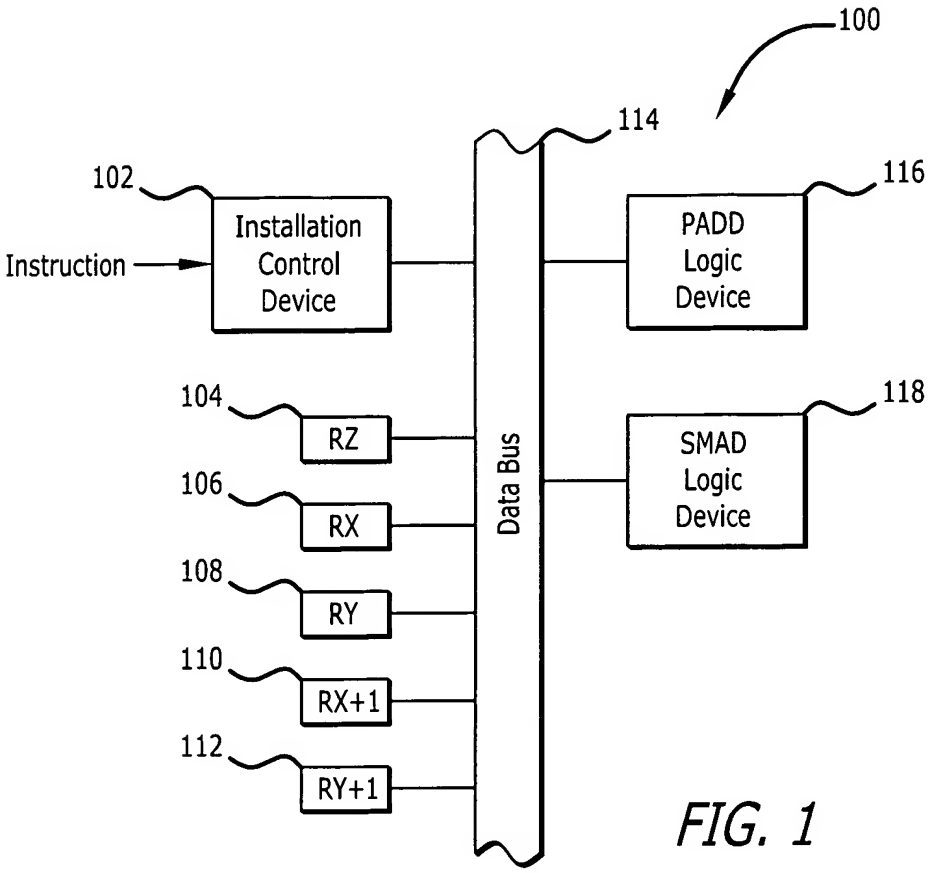


FIG. 1

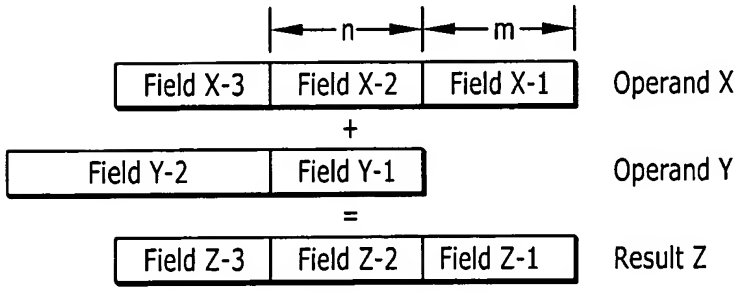


FIG. 4

200

<u>Instruction Description</u>

- PADD

Syntax: PADD [-C][-M] RZ, RX, RY
PADD -I[-C][-M] RZ, RX, <UI8 :immediate>
PADD -N[-C][-M] RZ, RX, RY, <UI5 : start>, <UI5 : stop>
PADD -N-I[-C][-M] RZ, RX, <UI8 : immediate>, <UI5 : start>

RX, RY are the source data registers
RZ is the destination register
<UI8: immediate> specifies the value of an immediate operand
<UI5: start> specifies start of bit field to be modified
<UI5: stop> specifies end of bit field to be modified
-C indicates addition with carry in
-M indicates addition modulo $2^n - 1$
-N indicates that addition affects only a bit field
-I indicates that second operand is supplied as an immediate value

FIG. 2A

250

Option Used	Operation
PADD RZ, RX, RY	RZ = RX + RY
PADD -C RZ, RX, RY	RZ = RX + RY + Cin
PADD -I RZ, RX, <U18: Immediate>	RZ = RX + <immediate>
PADD -N RZ, RX, RY, <UI5: Start>, <UI5: stop>	*RZ = { RX[31:stop], (RX[stop:start] + RY[length] + RX[start:0]) modulo 2 ^{length} }* Where length = stop - start + 1
PADD -M RZ, RX, RY	RZ = (RX + RY) modulo 2^{n}- 1
PADD -N -I RZ, RX, <UI8: immediate>, <UI5: start>	*RZ = { (RX[31:start], + immediate[31-start: 0]) modulo 2 $^{31-start+1}$, RX[start: 0] }* In this case, a stop is assumed to be 31.

FIG. 2B

SMAD

Syntax: SMAD [-A] [-M] RZ, RX, RY, <UI2: Length>, <UI2: Num Ops>

RZ is the destination register

RX and RY are source data registers

-A option is used to accumulate results where RZ is used as the accumulator

-M option results in a modulo 2^n- 1 addition

<UI2: Length> indicates the data widths

0: 8 bit operands, where each register is assumed to contain 4 8-bit operands

1: 16 bit operands, where each register is assumed to contain 2 16-bit operands

2: 32 bit operands

3: unused

<UI2: Num Ops> indicates the number of operands to be used in the addition

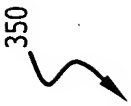
0: 2 source operands RX and RY

1: 3 source operands RX, RX+1 and RY

2: 3 source operands RX, RY and RY+1

3: 4 source operands RX, RY, RX+1 and RY+1

FIG. 3A



Option Used	Operation
SMAD RZ, RX, RY,2, 0	RZ = RZ + RX + RY
SMAD -A RZ, RX, RY, 2, 0	RZ = RX + RY + (RX+1) + (RY+1)
SMAD RZ, RX, RY, 2,3	*RZ = RX[7:0] + RX[15:8] + RX[23:16] + RX[31:24] + RY[7:0] + RY[15:8] + RY[23:16] + RY[31:24]*
SMAD RZ, RX, RY, 0, 0	RZ = (RX + RY) modulo 2^n - 1
SMAD -M RZ, RX, RY, 2, 0	RZ = (RZ + RX + RY) modulo 2^n - 1
SMAD -A -M RZ, RX, RY, 2, 0	

FIG. 3B

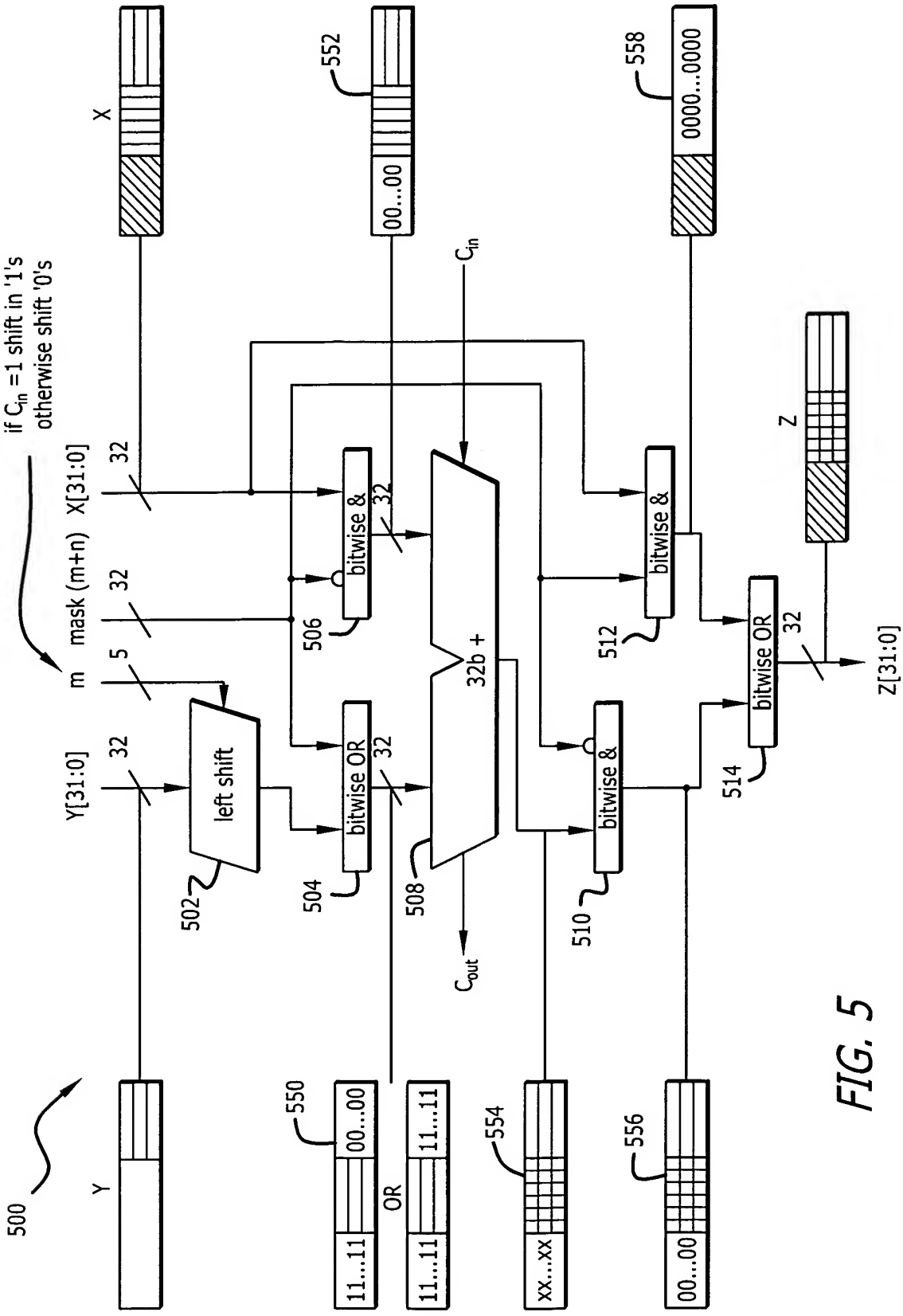

if C_{in} =1 shift in '1's
otherwise shift '0's
X
552
00...00
558
0000...0000
C_{in}
Z
X[31:0]
32
mask (m+n)
32
m
5
Y[31:0]
32
left shift
502
bitwise &
506
32
bitwise OR
504
32
32b +
508
C_{out}
bitwise &
512
bitwise &
510
bitwise OR
514
32
Z[31:0]
500
Y
550
00...00
11...11
OR
11...11
11...11
554
xx...xx
556
00...00

FIG. 5

m + n	mask (32b)	$\overline{\text{mask}}$ (32b)
0: 00000	1111...1110	0000...0001
1: 00001	1111...1100	0000...0011
2: 00010	1111...1000	0000...0111
⋮	⋮	⋮
30: 1_1110	1000...0000	0111...1111
31: 1_1111	0000...0000	1111...1111

FIG. 6

Carry bits of special consideration in "32-b CSA"			
carry bit	output from	input to	not propagated for modulo 2^n addition, n=?
c[8]	702-7	704-8	8
co_8	702-7	702-8	8
c[16]	702-15	704-0	8, 16
co_16	702-15	702-16	8, 16
c[24]	702-23	704-8	8
co_24	702-23	702-24	8
c[32]	702-31	708	8, 16, 32
co_32	702-31	702-0	8, 16, 32

FIG. 11

Carry bits of special consideration in "16-b CSA"			
carry bit	output from	input to	not propagated for modulo 2^naddition, n=? (n = 32 not applicable)
c[8]	704-7	706-0	8
co1_8	704-7	704-8	8
c1[16]	704-15	704-0	8, 16
co1_16	704-15	704-0	8, 16

FIG. 12

Carry bits of special consideration in "8-b CSA"			
carry bit	output from	input to	not propagated for modulo 2^n addition, n=? (n = 32, 16 not applicable)
c2[8]	706-7	706-0	8
co2_8	706-7	706-0	8

FIG. 13

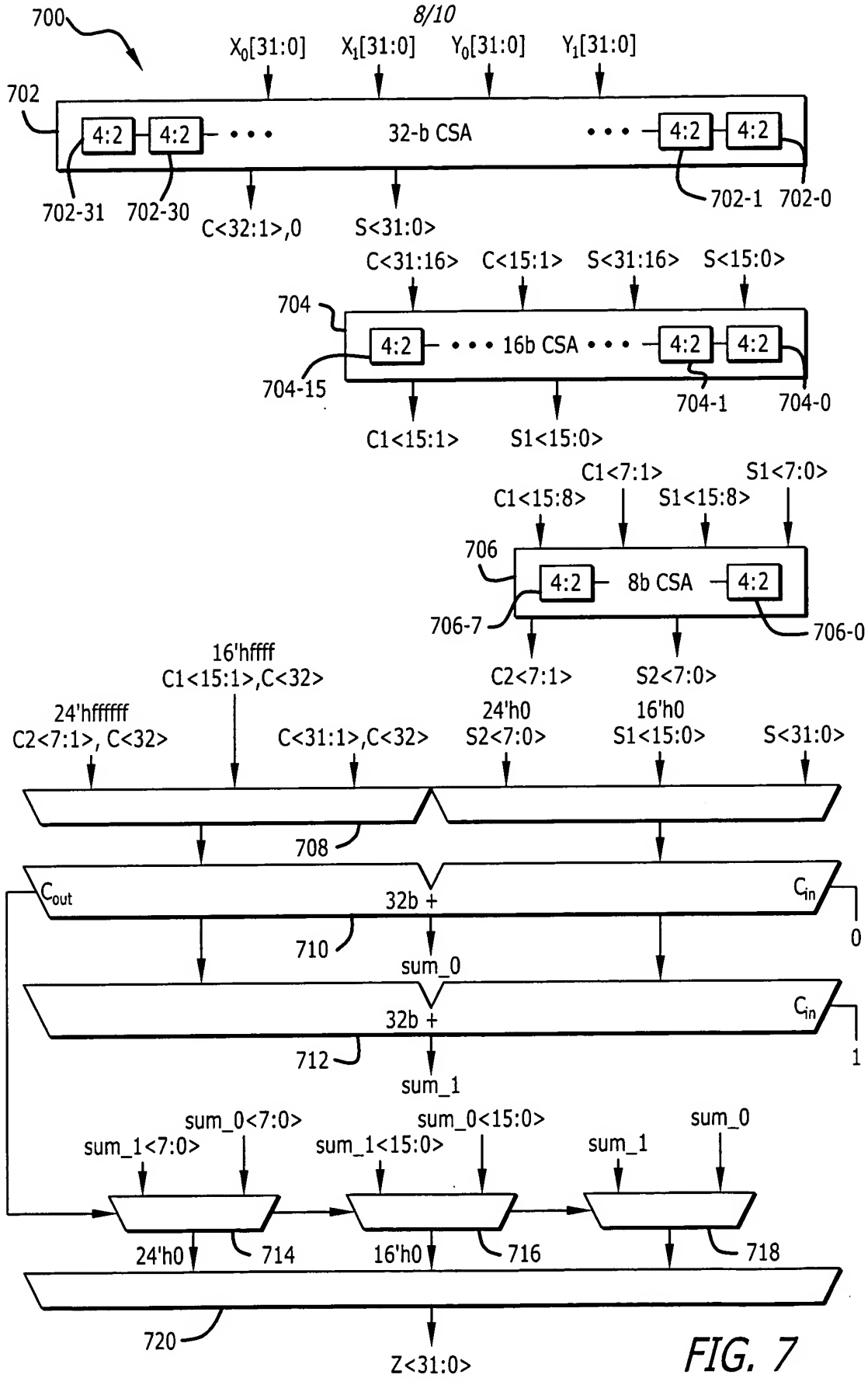

700
X0[31:0]
X1[31:0]
Y0[31:0]
Y1[31:0]
702
4:2
4:2
32-b CSA
4:2
4:2
702-31
702-30
C<32:1>,0
S<31:0>
702-1
702-0
C<31:16>
C<15:1>
S<31:16>
S<15:0>
704
4:2
16b CSA
4:2
4:2
704-15
704-1
704-0
C1<15:1>
S1<15:0>
C1<7:1>
S1<7:0>
C1<15:8>
S1<15:8>
706
4:2
8b CSA
4:2
706-7
706-0
C2<7:1>
S2<7:0>
16'hffff
C1<15:1>,C<32>
24'hffffff
C2<7:1>, C<32>
C<31:1>,C<32>
24'h0
S2<7:0>
16'h0
S1<15:0>
S<31:0>
708
Cout
32b +
Cin
0
710
sum_0
32b +
Cin
1
712
sum_1
sum_0<7:0>
sum_1<7:0>
sum_0<15:0>
sum_1<15:0>
sum_1
sum_0
24'h0
714
16'h0
716
718
720
Z<31:0>

FIG. 7

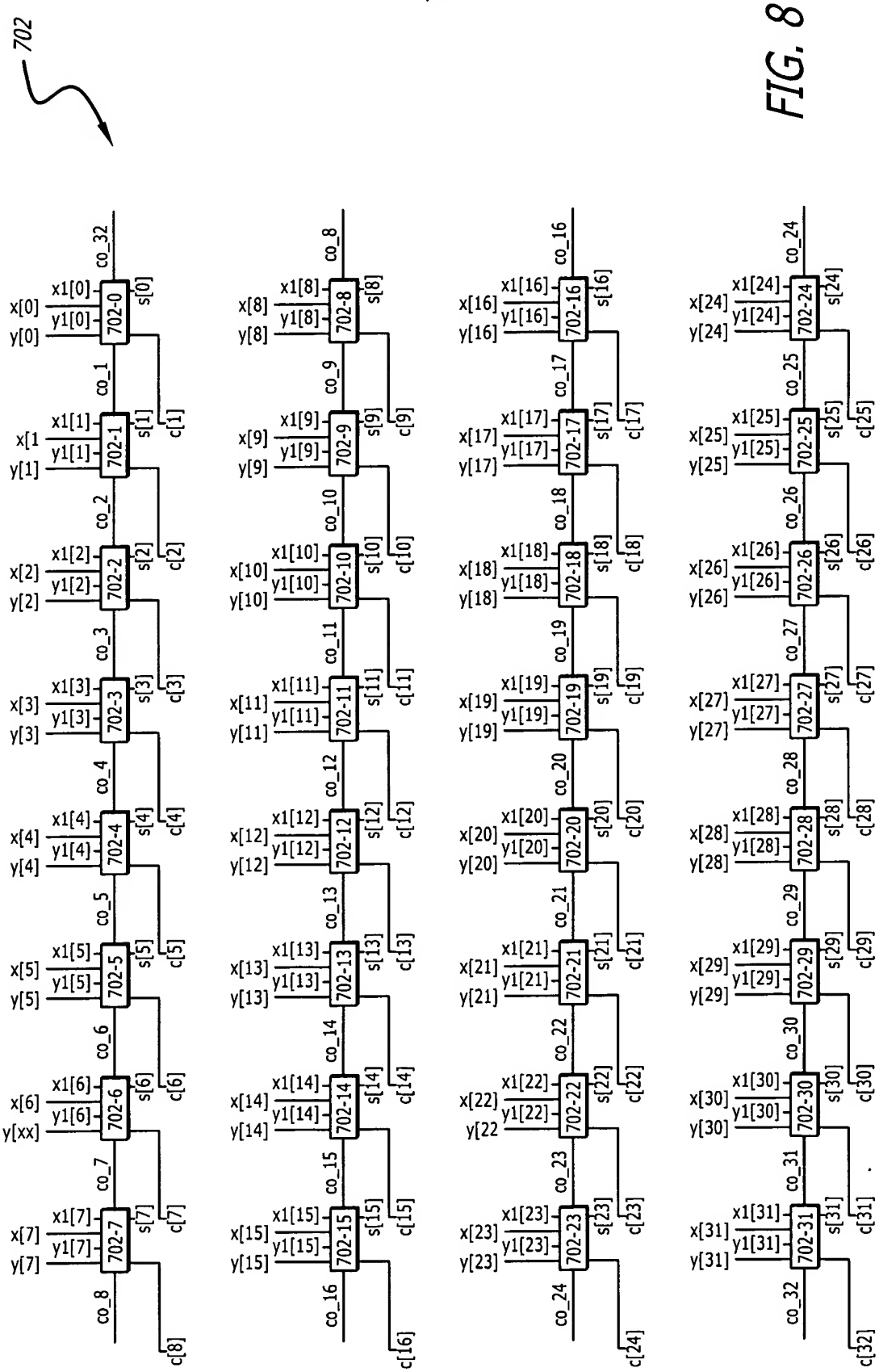

702
co_32
x1[0]
x[0]
y1[0]
y[0]
702-0
s[0]
co_1
x1[1]
x[1
y1[1]
y[1]
702-1
s[1]
c[1]
co_2
x1[2]
x[2]
y1[2]
y[2]
702-2
s[2]
c[2]
co_3
x1[3]
x[3]
y1[3]
y[3]
702-3
s[3]
c[3]
co_4
x1[4]
x[4]
y1[4]
y[4]
702-4
s[4]
c[4]
co_5
x1[5]
x[5]
y1[5]
y[5]
702-5
s[5]
c[5]
co_6
x1[6]
x[6]
y1[6]
y[xx]
702-6
s[6]
c[6]
co_7
x1[7]
x[7]
y1[7]
y[7]
702-7
s[7]
c[7]
co_8
c[8]
co_8
x1[8]
x[8]
y1[8]
y[8]
702-8
s[8]
co_9
x1[9]
x[9]
y1[9]
y[9]
702-9
s[9]
c[9]
co_10
x1[10]
x[10]
y1[10]
y[10]
702-10
s[10]
c[10]
co_11
x1[11]
x[11]
y1[11]
y[11]
702-11
s[11]
c[11]
co_12
x1[12]
x[12]
y1[12]
y[12]
702-12
s[12]
c[12]
co_13
x1[13]
x[13]
y1[13]
y[13]
702-13
s[13]
c[13]
co_14
x1[14]
x[14]
y1[14]
y[14]
702-14
s[14]
c[14]
co_15
x1[15]
x[15]
y1[15]
y[15]
702-15
s[15]
c[15]
co_16
c[16]
co_16
x1[16]
x[16]
y1[16]
y[16]
702-16
s[16]
co_17
x1[17]
x[17]
y1[17]
y[17]
702-17
s[17]
c[17]
co_18
x1[18]
x[18]
y1[18]
y[18]
702-18
s[18]
c[18]
co_19
x1[19]
x[19]
y1[19]
y[19]
702-19
s[19]
c[19]
co_20
x1[20]
x[20]
y1[20]
y[20]
702-20
s[20]
c[20]
co_21
x1[21]
x[21]
y1[21]
y[21]
702-21
s[21]
c[21]
co_22
x1[22]
x[22]
y1[22]
y[22
702-22
s[22]
c[22]
co_23
x1[23]
x[23]
y1[23]
y[23]
702-23
s[23]
c[23]
co_24
c[24]
co_24
x1[24]
x[24]
y1[24]
y[24]
702-24
s[24]
co_25
x1[25]
x[25]
y1[25]
y[25]
702-25
s[25]
c[25]
co_26
x1[26]
x[26]
y1[26]
y[26]
702-26
s[26]
c[26]
co_27
x1[27]
x[27]
y1[27]
y[27]
702-27
s[27]
c[27]
co_28
x1[28]
x[28]
y1[28]
y[28]
702-28
s[28]
c[28]
co_29
x1[29]
x[29]
y1[29]
y[29]
702-29
s[29]
c[29]
co_30
x1[30]
x[30]
y1[30]
y[30]
702-30
s[30]
c[30]
co_31
x1[31]
x[31]
y1[31]
y[31]
702-31
s[31]
c[31]
co_32
c[32]

FIG. 8

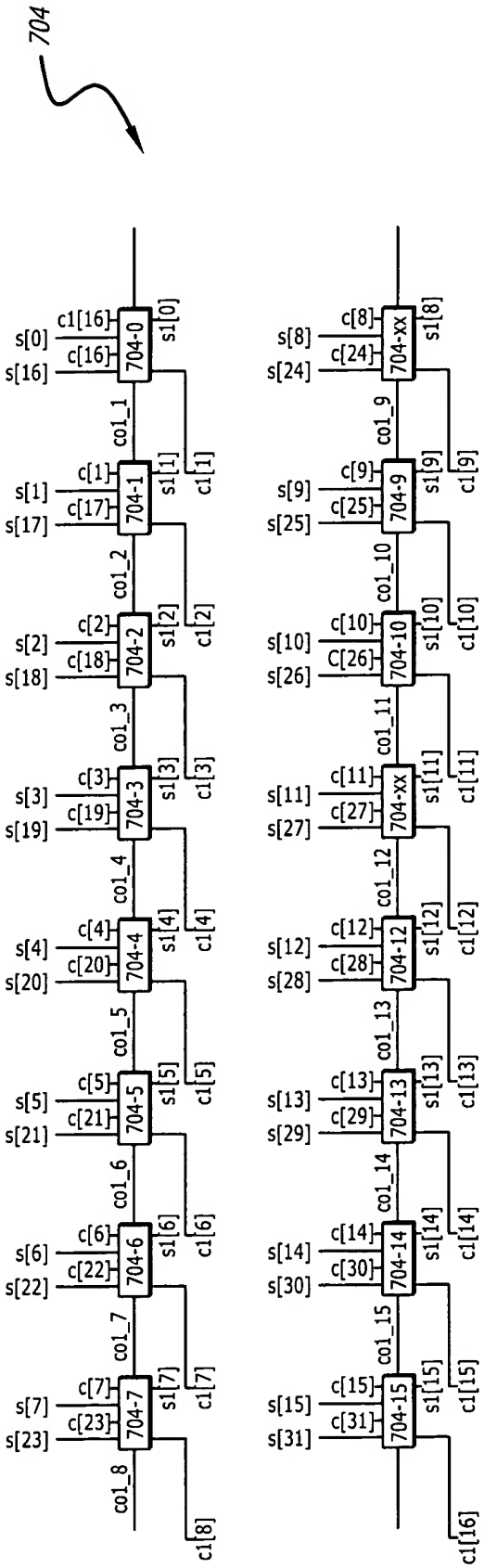

704
s[0]
s[16]
c1[16]
c[16]
704-0
s1[0]
co1_1
s[1]
s[17]
c[1]
c[17]
704-1
s1[1]
c1[1]
co1_2
s[2]
s[18]
c[2]
c[18]
704-2
s1[2]
c1[2]
co1_3
s[3]
s[19]
c[3]
c[19]
704-3
s1[3]
c1[3]
co1_4
s[4]
s[20]
c[4]
c[20]
704-4
s1[4]
c1[4]
co1_5
s[5]
s[21]
c[5]
c[21]
704-5
s1[5]
c1[5]
co1_6
s[6]
s[22]
c[6]
c[22]
704-6
s1[6]
c1[6]
co1_7
s[7]
s[23]
c[7]
c[23]
704-7
s1[7]
c1[7]
co1_8
c1[8]
s[8]
s[24]
c[8]
c[24]
704-xx
s1[8]
co1_9
s[9]
s[25]
c[9]
c[25]
704-9
s1[9]
c1[9]
co1_10
s[10]
s[26]
c[10]
C[26]
704-10
s1[10]
c1[10]
co1_11
s[11]
s[27]
c[11]
c[27]
704-xx
s1[11]
c1[11]
co1_12
s[12]
s[28]
c[12]
c[28]
704-12
s1[12]
c1[12]
co1_13
s[13]
s[29]
c[13]
c[29]
704-13
s1[13]
c1[13]
co1_14
s[14]
s[30]
c[14]
c[30]
704-14
s1[14]
c1[14]
co1_15
s[15]
s[31]
c[15]
c[31]
704-15
s1[15]
c1[15]
c1[16]

FIG. 9

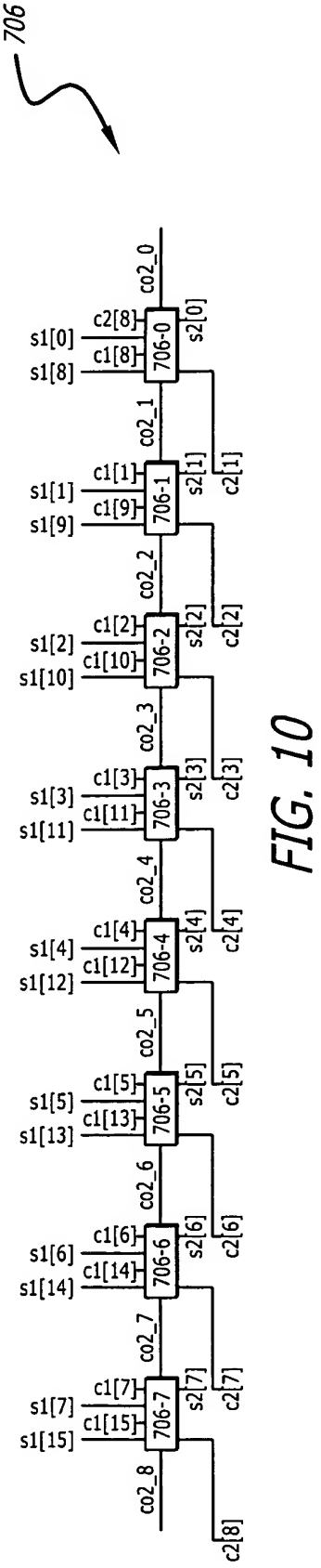

706
s1[0]
s1[8]
c2[8]
c1[8]
706-0
s2[0]
co2_0
co2_1
s1[1]
s1[9]
c1[1]
c1[9]
706-1
s2[1]
c2[1]
co2_2
s1[2]
s1[10]
c1[2]
c1[10]
706-2
s2[2]
c2[2]
co2_3
s1[3]
s1[11]
c1[3]
c1[11]
706-3
s2[3]
c2[3]
co2_4
s1[4]
s1[12]
c1[4]
c1[12]
706-4
s2[4]
c2[4]
co2_5
s1[5]
s1[13]
c1[5]
c1[13]
706-5
s2[5]
c2[5]
co2_6
s1[6]
s1[14]
c1[6]
c1[14]
706-6
s2[6]
c2[6]
co2_7
s1[7]
s1[15]
c1[7]
c1[15]
706-7
s2[7]
c2[7]
co2_8
c2[8]

FIG. 10